

What Is Claimed Is:

1. A servo loop control apparatus comprising:  
input circuitry arranged to receive signals from  
a servo loop to be controlled;  
output circuitry arranged to provide signals to  
a servo loop to be controlled;  
a master processor to control said apparatus,  
said master processor being connected to an input/output  
bus arbiter;  
a second processor, said second processor  
programmed to operate autonomously of said master  
processor and being dedicated to specific, pre-programmed  
servo loop control tasks and being connected to said  
input/output bus arbiter;  
said input/output bus arbiter being configured  
to control access to said input circuitry and said output  
circuitry by said master processor and said second  
processor.
2. The apparatus recited in claim 1, wherein said  
second processor comprises means for executing repetitive  
preprogrammed servo loop processing instructions without  
interruption, thereby maintaining operation of a servo  
loop independent of said master program.
3. The apparatus recited in claim 2, further  
comprising a plurality of vector registers, said vector  
registers forming a means for interrupting said servo loop  
processing in said second processor under a limited set  
of predefined conditions.
4. The apparatus recited in claim 3, further  
comprising an instruction RAM for access by said second  
processor, said instruction RAM storing instructions for  
user defined servo control loop routines.

5. The apparatus recited in claim 4, further comprising a memory accessible by said second processor while executing said instructions, said memory storing for access at least one of a subroutine and a variable parameter used by said user defined servo control loop routines.

6. The apparatus recited in claim 3, further comprising an instruction ROM for access by said second processor, said instruction ROM storing instructions for predefined turn-key routines and used controlling a servo loop.

7. The apparatus recited in claim 5, further comprising a memory accessible by said second processor while executing said instructions, said memory storing for access at least one of a subroutine and a variable parameter used by said turn-key routines.

8. The apparatus recited in claim 2, further comprising a third processor, said third processor being interruptable to handle asynchronous servo loop events.

9. The apparatus recited in claim 7, wherein said bus arbiter comprises programmable means for allocating priority of access to said input and output circuitry to said second processor, thereby preserving synchronicity of said second processor task.

10. The apparatus recited in claim 1, wherein said input circuitry comprises a multiplexer receiving inputs on a plurality of separate channels, said multiplexer having an output periodically outputting a signal from each channel in a predetermined sequence, thereby maintaining freshly updated data for each channel on said output.

11. The apparatus recited in claim 9 wherein said multiplexer has means responsive to said second processor for interrupting said predetermined sequence and for outputting a signal for a particular channel requested by said second processor.

12. A method of servo loop control, the method comprising the steps of:

receiving into input circuits signals from a servo loop to be controlled;

through output circuitry providing signals to a servo loop to be controlled;

executing overall control of said servo loop with a master processor connected to an input/output bus arbiter;

operating a second processor, connected to said input/output bus arbiter, autonomously of said master processor and dedicating said second processor to specific, pre-programmed servo loop control tasks; and

controlling access to said input circuitry and said output circuitry by said master processor and said second processor with said input/output bus arbiter.

13. The method recited in claim 12, further comprising executing in said second processor repetitive preprogrammed servo loop processing instructions without interruption, thereby maintaining operation of a servo loop independent of said master program.

14. The method recited in claim 13, further comprising interrupting said servo loop processing in said second processor under a limited set of predefined conditions through a plurality of vector registers.

15. The method recited in claim 14, further comprising storing instructions for user defined servo control loop routines in an instruction RAM for access by said second processor.

16. The method recited in claim 15, further comprising storing for access at least one of a subroutine and a variable parameter used by said user defined servo control loop routines in a memory accessible by said second processor while executing instructions.

17. The method recited in claim 14, further comprising storing instructions for predefined turn-key routines used in controlling a servo loop in an instruction ROM for access by said second processor.

18. The method recited in claim 17, further comprising storing for access at least one of a subroutine and a variable parameter used by said turn-key routines in a memory accessible by said second processor while executing said instructions.

19. The method recited in claim 13, further comprising processing asynchronous servo loop events with a third processor interruptable to handle said asynchronous servo loop events.

20. The method recited in claim 19, comprising allocating, through said bus arbiter, priority of access to said input and output circuitry to said second processor, thereby preserving synchronicity of said second processor task.

21. The method recited in claim 12, comprising receiving inputs on a plurality of separate channels of a multiplexer and periodically outputting and accessing a signal from each channel in a predetermined sequence, thereby maintaining freshly updated data for each channel on said output.

22. The method recited in claim 21, wherein responsive to said second processor, said multiplexer interrupts said predetermined sequence and outputs a

signal for a ~~B~~ particular channel requested by said second processor.

Add B17

Add  
E3 /